

7. Kimberley

Program Name: Kimberley.java

Input File: kimberley.dat

Kimberley loves watching birds, especially the large flocks that move together in flight so gracefully. She has recently run across a theory study of how these flocks move, and needs your help in simulating a small part of that study.

Using a single row of birds that always fly from left to right, she wants to see how some simple rules will determine the position of each bird in the row after a certain time stage as the birds fly.

The rules of the simulation are:

1. At the beginning of any stage change, if the left most space in the row is empty and the right most space is occupied by a bird, that bird leaves the right most space empty and moves to the left most space for the next stage.
2. A bird stays in place if the space to the right of it is occupied.
3. A bird moves one place to the right if that space is empty.
4. A bird moves at most one time during a stage change.

For example, the diagram shown below represents a flock of birds in three stages, moving according to the rules listed above.

Position	1	2	3	4	5	6	7	8	9	10
Stage 1	*		*	*		*		*	*	*
Stage 2		*	*		*		*	*	*	*
Stage 3	*	*		*		*	*	*		*

From stage 1 to 2, the birds in positions 1, 4, and 6 move. All others remain.

From stage 2 to 3, the bird in space 10 falls back to position 1, and then the birds in 3, 5 and 9 all move.

Input: Several sets of data, each data set consisting of a row of integers with single space separation. The first integer represents the number of spaces to be used in the row. The next integer N indicates how many birds are in the row, with the next N values indicating their positions in the row. The final integer represents the number of stages to be considered for the simulation.

Output: After the final stage of each data set is completed, report the positions of all of the birds as a row of integers with single space separation.

Sample input:

```
9 3 1 3 5 2
12 6 1 3 5 7 10 11 4
10 7 1 3 4 6 8 9 10 3 (this data set matches the example described above)
```

Sample output:

```
2 4 6
2 4 6 8 10 12
1 2 4 6 7 8 10
```